

VBA: Useful Code for Multi-Scale Caching

Contributed by Bert Granberg
01, Dec. 2008
Last Updated 01, Dec. 2008

This page will be used as a repository for VBA code that was written to help make the process of developing an ArcMap MXD for ArcGIS Server map caching easier.

The functionality includes:

- `renameLayersInGroupLayerViaSearchAndReplace()` allows you to select a group layer (only works on group layers) in the Table of Contents and then do a search and replace on two levels of 'child' layers. When making multi-scale map projects, it's often useful to make a set of layers applying to one scale range, give them an ID, then copy them and rename and resymbolize for another scale range.
- `rePointSDELayersToAppConnORDirectConnORFileGDB()` allows you to repoint feature classes from SDE connections to local fileGDBs. When creating an ArcGIS Server map cache, it is much more efficient to copy your SDE feature classes to File Geodatabase format on the caching machine. If you keep the feature class names the same, this script will repoint all the layers to the local fileGDB location. The script can also be used to change SDE Application connects (app connects) to SDE direct connections and vice versa. NOTE: you will need to set some parameters in this sub procedure and also the called function...look for the *****s

'updated 12/01/08 1:52 pm

```
Public Sub renameLayersInGroupLayerViaSearchAndReplace()
```

```
    Dim findStr As String
    Dim replaceStr As String
```

```
    findStr = "Scale_01"
    replaceStr = "Scale_02"
```

```
    Dim pMXD As IMxDocument
    Dim pMap As IMap
```

```
    Dim pCompLayer As ICompositeLayer 'group layer
    Dim pLayer As ILayer
    Dim c As Integer
```

```
    Set pMXD = ThisDocument
    Set pMap = pMXD.FocusMap
    Set pCompLayer = pMXD.SelectedLayer
```

```
    For c = 0 To pCompLayer.Count - 1
        Set pLayer = pCompLayer.Layer(c)
        pLayer.Name = Replace(pLayer.Name, findStr, replaceStr)
    Next
```

```
End Sub
```

```
Public Sub rePointSDELayersToAppConnORDirectConnORFileGDB()
```

```
    Dim pMXD As IMxDocument
    Dim pMap As IMap
    Dim gl As ILayer
    Dim fl As IFeatureLayer
    Dim cl As ICompositeLayer
    Dim cl2 As ICompositeLayer
    Dim c As Integer
    Dim c2 As Integer
    Dim c3 As Integer
    Dim fc As IFeatureClass
```

```

Dim ds As IDataset
Dim l As ILayer
Dim grpLayerIndexNumber As Integer
Dim changeType As Integer

```

```

'***** SET THIS PARAMETER
'changeType = 1 'change data source to SDE Direct Connect
'changeType = 2 'change data source to SDE Application Connect
changeType = 3 'change data source to file Geodatabase
'*****

```

```

Set pMXD = ThisDocument
Set pMap = pMXD.FocusMap 'work on active dataframe only

```

```

For c = 0 To pMap.LayerCount - 1
    Debug.Print pMap.Layer(c).Name
    If TypeOf pMap.Layer(c) Is ICompositeLayer Then
        Set cL = pMap.Layer(c)
        For c2 = 0 To cL.Count - 1
            If TypeOf cL.Layer(c2) Is ICompositeLayer Then
                Set cL2 = cL.Layer(c2)
                For c3 = 0 To cL2.Count - 1
                    If TypeOf cL2.Layer(c3) Is IFeatureLayer Then
                        Set fL = cL2.Layer(c3)
                        Set fL.FeatureClass = changeFCWS(fL.FeatureClass, changeType)
                    End If
                Next c3
            End If
            If TypeOf cL.Layer(c2) Is IFeatureLayer Then
                Set fL = cL.Layer(c2)
                Set fL.FeatureClass = changeFCWS(fL.FeatureClass, changeType)
            End If
        Next c2
    End If
    If TypeOf pMap.Layer(c) Is IFeatureLayer Then
        Set fL = pMap.Layer(c)
        Set fL.FeatureClass = changeFCWS(fL.FeatureClass, changeType)
    End If
Next c

End Sub

```

```

Public Function changeFCWS(inFC As IFeatureClass, inType As Integer) As IFeatureClass

```

```

    Dim fileGDBPath As String
    Dim ds As IDataset
    Dim n As String
    Dim sdeApp_WF As IWorkspaceFactory
    Dim sdeDC_WF As IWorkspaceFactory
    Dim FGDB_WF As IWorkspaceFactory
    Dim pWS As IFeatureWorkspace
    Dim pPropSet As IPropertySet
    Dim databaseDotOwnerStr As String
    Dim databaseConnUserNameStr As String
    Dim databaseConnPasswordStr As String
    Dim databaseConnPortStr As String
    Dim directConnectInstanceStr as String

```

```

'***** SET THESE VALUES *****
databaseDotOwnerStr = "SGID.U024."
databaseConnUserNameStr = "agrc"
databaseConnPasswordStr = "ENTER PASSWORD HERE"
databaseConnPortStr = "ENTER PORT HERE"
fileGDBPath = "c:\supercache.gdb"

```

```
directConnectInstanceStr = "192.000.000\InstanceName"
```

```
*****
```

```
Set ds = inFC
```

```
n = ds.Name
```

```
Debug.Print " " & n
```

```
If inType = 1 Then
```

```
    'Repoint to DirectConnect
```

```
    Set pPropSet = New PropertySet
```

```
    With pPropSet
```

```
        .SetProperty "Server", "direct connect"
```

```
        .SetProperty "Instance", "sde:sqlserver:" & directConnectInstanceStr
```

```
        .SetProperty "user", databaseConnUserNameStr
```

```
        .SetProperty "database", "sgid"
```

```
        .SetProperty "password", databaseConnPasswordStr
```

```
        .SetProperty "version", "sde.default"
```

```
    End With
```

```
    Set sdeDC_WF = New SdeWorkspaceFactory
```

```
    Set pWS = sdeDC_WF.Open(pPropSet, 0)
```

```
Elseif inType = 2 Then
```

```
    'Repoint to App Connect
```

```
    Set pPropSet = New PropertySet
```

```
    With pPropSet
```

```
        .SetProperty "Server", "agrc.state.ut.us"
```

```
        .SetProperty "Instance", databaseConnPortStr
```

```
        .SetProperty "user", databaseConnUserNameStr
```

```
        .SetProperty "database", "sgid"
```

```
        .SetProperty "password", databaseConnPasswordStr
```

```
        .SetProperty "version", "sde.default"
```

```
    End With
```

```
    Set sdeDC_WF = New SdeWorkspaceFactory
```

```
    Set pWS = sdeDC_WF.Open(pPropSet, 0)
```

```
Elseif inType = 3 Then
```

```
    'Repoint to FGDB
```

```
    Set FGDB_WF = New FileGDBWorkspaceFactory
```

```
    Set pWS = FGDB_WF.OpenFromFile(fileGDBPath, 0)
```

```
End If
```

```
'open feature class at new location
```

```
If inType = 1 Or inType = 2 Then
```

```
    If InStr(n, databaseDotOwnerStr) = 0 Then
```

```
        Set changeFCWS = pWS.OpenFeatureClass (databaseDotOwnerStr & n)
```

```
    Else
```

```
        Set changeFCWS = pWS.OpenFeatureClass (n)
```

```
    End If
```

```
Elseif inType = 3 Then
```

```
    Set changeFCWS = pWS.OpenFeatureClass(Replace(n, databaseDotOwnerStr, ""))
```

```
End If
```

```
End Function
```